COMPSCI 230: Discrete Mathematics for Computer Science	March 18, 2019
Lecture 16	
Lecturer: Debmalya Panigrahi	*Scribe: Erin Taylor*

1 Overview

In this lecture, we continue our study of graph theory. We introduce properties related to graph connectivity. We also introduce the class of acyclic graphs and discuss trees, which have many applications in computer science.

2 Connectivity

Throughout this section, we let $G = (V, E)$ be a graph where $|E| = m$, $|V| = n$. Last lecture, we defined connectivity as an equivalence relation on the vertices. Informally, recall that we say two vertices u and v are *connected* if there is a path that begins at u and ends at v. A *connected component* of a graph is a subgraph consisting of a vertex u and all vertices connected to u, along with any edges incident on these vertices. See Figure 1 for an example of a graph with 3 connected components. We say a graph is *connected* if it has only one connected component.

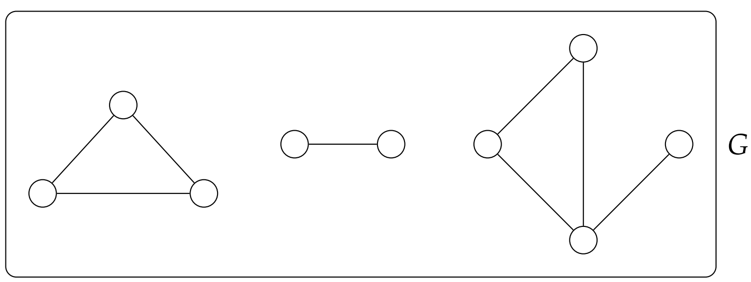

Figure 1: A graph with 3 connected components.

Definition 1. *An edge e is a* cut edge *if the removal of e increases the number of connected components.*

Claim 1. *Edge e is a* cut edge *if and only if it is not contained in any cycle.*

Proof. We will prove the equivalence between the two characterizations of a cut edges by proving the contrapositive of Claim 1. Definition 1 implies that the removal of a cut edge disconnects two vertices that are connected in G. If an edge $e = (u, v)$ is not a cut edge, then after removing e there must still be a path from u to v. Joining this path with e must create a cycle. Conversely, an edge that is on a cycle cannot be a cut edge by definition. $\square$

Theorem 2. *The number of connected components in a graph is* $\geq n - m$.

Proof. We will prove Theorem 2 by induction on the number of edges of G. Let $G = (V, E)$ be any graph with n vertices.

Base Case: If $m = 0$, then G is an empty graph. Each vertex forms its own connected component. Thus, the number of connected components is n, and $n \geq n - 0$.

Inductive Hypothesis: For any graph G' with $m' < m$ edges, the number of connected components of G' is $\geq n - m'$.

Inductive Step: Let G be an arbitrary n-vertex graph with m edges. Let k be the number of connected components of G. Remove an arbitrary edge $e = (u, v)$. Let G' be the resulting graph: $G' = (V, E \setminus \{e\})$. There are two cases to consider.

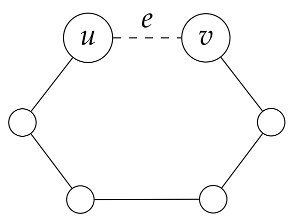

Figure 2: Example of Case 1: There is an alternative path from u to v in graph G.

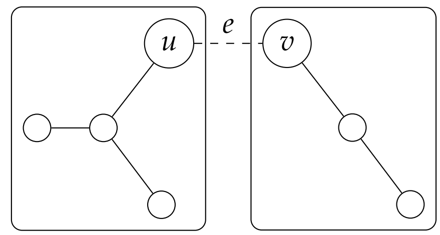

Figure 3: Example of Case 2: There is no alternative path from u to v in graph G. Vertices u and v become disconnected.

Case 1: e is not a cut edge. Thus, there exists a path from u to v which does not include e. See Figure 2 for an example. Call one such path p. Let's consider how connectedness changes in this case. Suppose s, t are two vertices in G.

1. If s and t were not connected in G, then they remain disconnected in G' since removing an edge cannot make a previously disconnected pair of vertices connected.

2. If s, t were connected via a path that did not contain e, this path was not changed in G', so they remain connected.

3. Finally, suppose s, t were connected by a path which includes e in G. Replace edge e in the path by inserting path p. This creates a walk from s to t. Recall our proof that if there exists a walk from s to t, there must exist a walk from s to t (you can find this path by deleting repeated portions of the walk). Thus, there still exists a valid path from from s to t, so they remain connected.

Thus, the connection status does not change for any pair of vertices. Therefore, the number of connected components in G' is the same as in G.

Case 2: e is a cut edge. Thus, there does not exist an alternative path from u to v. See Figure 3 for an example. In this case, the number of connected components increases by 1. Suppose s, t are two vertices in G. Their connectedness only changes if they were connected by a path which included e. Since there is no alternative path, s and t will become disconnected. One of s, t will be connected to u, and the other to v. Thus, the number of connected components in G' is one more than the number in G.

Let k' be the number of connected components in G'. After removing e, the number of connected components either stays the same or increases by 1. In both cases, $k' \leq k + 1$. Since G' has $m - 1$

edges, we can apply the inductive hypothesis. Thus, $k + 1 \geq k' \geq n - (m - 1)$. This implies $k \geq n - m$, as desired. □

Corollary 3. *Any connected graph has $m \geq n - 1$.*

Proof. In a connected graph, the number of connected components is 1. Plugging 1 into Theorem 2 immediately gives the corollary: $1 \geq n - m \Leftrightarrow m \geq n - 1$. □

Next, we introduce a common class of graphs, and study properties of connectivity for this class.

3 Acyclic Graphs

3.1 Forests and Trees

Definition 2. *A graph is* acyclic *if it does not contain a cycle. Acyclic graphs are also called* forests*.*

Theorem 4. *Any acyclic graph has $m \leq n - 1$.*

In an acyclic graph, all edges are cut edges.

Proof of Theorem 4. We will proceed via induction on the number of edges of G. Let $G = (V, E)$ be any graph with n vertices.
Base Case: If $m = 0$, then $0 \leq n - 1$ for any $n \geq 1$.
Inductive Hypothesis: Let m be an arbitrary positive integer. For any acyclic graph G' with $m' < m$ edges, the number of edges of G' is $m' \leq n - 1$.
Inductive Step: Let G be an arbitrary n-vertex graph with m edges. Remove an edge $e = (u, v)$ from G. Since G is acyclic, e is necessarily a cut edge. Thus, the number of components increases by one after removing e. Let G' be the resulting graph after removing e. Consider two subgraphs of G'. Let S_1 be the subgraph containing all connected components of G' that do not contain v, and let S_2 be the subgraph that is just the connected component containing v. Suppose S_1 has n_1 vertices and m_1 edges, and S_2 has n_2 vertices and m_2 edges. By construction $n_1 + n_2 = n$ and $m_1 + m_2 = m - 1$. By the inductive hypothesis on S_1 and S_2, $m_1 \leq n_1 - 1$ and $m_2 \leq n_2 - 1$. Putting these facts together:

$$m - 1 = m_1 + m_2 \leq (n_1 - 1) + (n_2 - 1) = (n_1 + n_2) - 2 = n - 2 \Rightarrow m \leq n - 1.$$

Thus, the theorem holds for all acyclic graphs. □

Definition 3. *A* tree *is a connected acyclic graph.*

Lemma 5. *A tree has $n - 1$ edges.*

Proof. Suppose we have a tree. By definition 3, this means the graph is connected and acyclic. By Corollary 3, $m \geq n - 1$. By Theorem 4, $m \leq n - 1$. Thus, $m = n - 1$, as desired. □

Theorem 6. *The following are three properties of a tree:*

1. *A tree is connected.*
2. *A tree is acyclic.*

3. *A tree has m = n - 1.*

Any two of these properties imply the third.

This theorem implies that any two properties (connected, acyclic, $n-1$) can be used to define a tree. If any two properties hold, they immediately imply that the third also holds.

Proof of Theorem 6. $\mathbf{(1)} \wedge \mathbf{(2)} \rightarrow \mathbf{(3)}$: This is Lemma 5.

$\mathbf{(1)} \wedge \mathbf{(3)} \rightarrow \mathbf{(2)}$: A connected graph with $n-1$ edges is acyclic.

We will proceed with a proof by contradiction. Suppose G is connected, has $n-1$ edges, and has at least one cycle. Remove an edge from a cycle in G, and G remains connected. Thus, we have a connected graph with one less edge: $m = n-2$. This contradicts Theorem 2. Thus, G must have been acyclic, as desired.

$\mathbf{(2)} \wedge \mathbf{(3)} \rightarrow \mathbf{(1)}$: An acyclic graph with $n-1$ edges is connected.

We will proceed with a proof by contradiction. Suppose G is acyclic, has $n-1$ edges, and is disconnected. G consists of two or more components, and each component is acyclic. Consider two connected components of G. Let the vertices of the two components be C_1 and C_2. Pick a $u \in C_1$ and $v \in C_2$ and add an edge between vertex u and v. There was no path between u and v in G since they were in different connected components, so we did not create a cycle. After adding edge (u, v), G is a graph of n vertices, n edges, and no cycles. This contradicts Theorem 4. Thus, G must have been connected, as desired. □

3.2 Spanning Trees

Definition 4. *A* spanning tree *of a connected graph $G = (V, E)$ is a subgraph G that is a tree on the set of vertices V.*

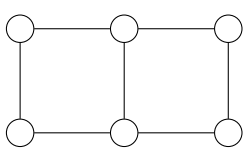

Figure 4: A graph G.

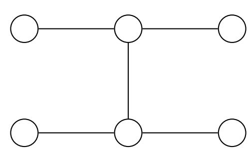

Figure 5: A spanning tree of G in Figure 4.

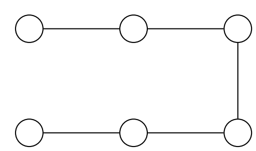

Figure 6: Another spanning tree of G in Figure 4.

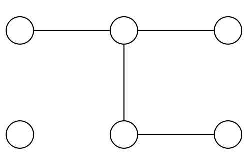

Figure 7: This subgraph does not connect all vertices of G in Figure 4.

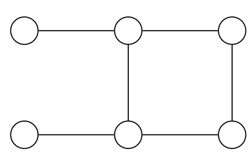

Figure 8: This subgraph of G in Figure 4 is not a tree.

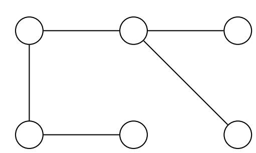

Figure 9: Not a valid subgraph of G in Figure 4.

Note that spanning trees are not necessarily unique. Suppose we want to find a spanning tree of graph G in Figure 4. Both Figure 5 and Figure 6 are valid spanning trees of G. The definition specifies that a spanning tree must connect all vertices of G, the subgraph in Figure 7 does not satisfy this condition. The subgraph in Figure 8 is not a tree, as it contains a cycle. The graph in Figure 9 uses an edge that was not originally in E, so it is not a valid subgraph. All three conditions of Definition 4 must be satisfied to have a valid spanning tree.

Observe that if the graph is a tree, there is one unique spanning tree and it is the graph itself. Spanning trees are especially important for graphs whose edges are assigned numerical weights.

Definition 5. *A* weighted graph $G = (V, E)$ *is one in which each edge is associated with a real number* $\boldsymbol{w}(e)$*, called its weight.*

Definition 6. *The* weight *of a graph* $G = (V, E)$ *is the sum of the weights of all its edges:*

$$\boldsymbol{weight}(G) = \sum_{e \in E} \boldsymbol{w}(e).$$

Given a weighted graph, we may be interested in finding a spanning tree with the minimum weight, called a *minimum spanning tree* (MST).

Lemma 7. *Given a weighted graph* $G = (V, E)$ *with edge weights* $w(e)$*, MST* T^**, and a constant* $c \in \mathbb{R}$*, define new edge weights as follows:*

$$\forall e \in E.\ w'(e) = w(e) + c.$$

T^* *remains an MST of* G*.*

Proof. Consider an arbitrary spanning tree T of weighted graph $G = (V, E)$. We will compute the change in cost of T after the weight shift. Let $\boldsymbol{weight}(T)$ be the original weight of spanning tree T and $\boldsymbol{weight}'(T)$ be the new weight of T. Then,

$$\boldsymbol{weight}'(T) = \sum_{e \in T} (w(e) + c) = \sum_{e \in T} w(e) + (n-1)c = \boldsymbol{weight}(T) + (n-1)c.$$

Thus, the weight of all spanning trees increases by the same amount. This is a consequence of the fact that there are always $n - 1$ edges in a spanning tree. Therefore, the minimum spanning tree still has the minimum weight after this constant weight shift of all edges. □

Lemma 7 is a useful property for finding a MST and related problems. While we will not study the problem of finding a minimum spanning tree in depth, we will explore properties of the MST.

In the rest of the section, we assume that all edge weights are distinct. This ensures that the MST of a graph is unique. This assumption is essentially without loss of generality, as there is a scheme to ensure distinctness that does not change the identity of the MST. We do not describe this process, but assume it can be done. Now, we will introduce a few definitions in order to prove properties about the minimum spanning tree.

Definition 7. *A* black-white coloring *of a graph* G *is a partition of the vertices into two sets. All vertices in one set are colored white, and all vertices in the other set are colored black. A* gray edge *of a black-white coloring is an edge with different colored endpoints.*

We have the following properties of an MST:

1. **Cycle Property:** In any cycle, the maximum weight edge will not appear in the MST.

2. **Cut Property:** If the MST is unique, for any black-white coloring of G, the MST will contain the minimum weight gray edge. In the case that the edge weights are not distinct, for any black-white coloring of G, the minimum weight gray edge will be contained in some MST.

The second property is somewhat surprising. There are 2^n different black-white colorings of a graph G (for each vertex, there are 2 choices for its color), but an MST only has $n-1$ edges. This implies many of the minimum-weight gray edges are in different black-white colorings are the same edge.

Both properties immediately yield methods for finding minimum spanning trees. Consider the cycle property. Given a graph, first we find a cycle. We can remove the maximum weight edge from that cycle. By the Cycle Property, this edge could not be in the MST. We can repeat this process of finding a cycle and removing an edge until no cycles are left. The remaining edges form the MST.

Well-known algorithms for finding the MST exploit the cut property. Here is one such method for incrementally building the MST:

1. Pick an arbitrary vertex, color it black. Color all other vertices white.
2. Find the minimum weight gray edge and add it to the tree we are building.
3. Color the other vertex incident on the chosen edge black. Recolor edges appropriately.
4. Repeat Steps 2 and 3 until all vertices are black.

See Figure 10 and Figure 11 for an example of this process. We start with one black vertex and pick the minimum weight gray edge (subsequently colored in blue), resulting in two black vertices. We continue this process until all vertices are black, and the blue edges make up the MST.

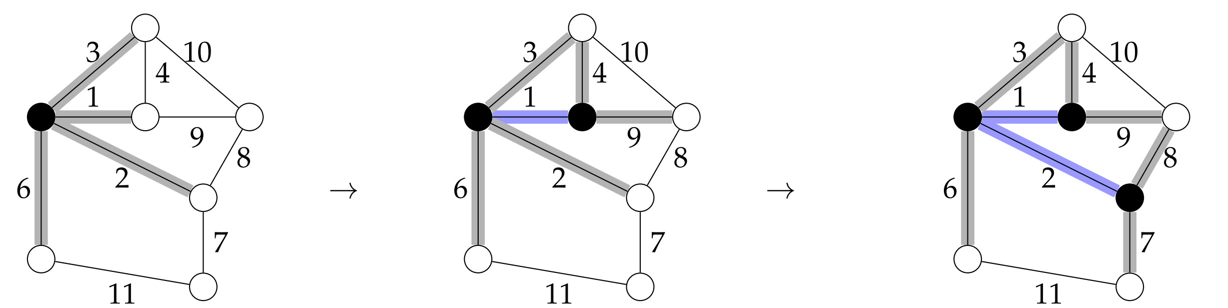

Figure 10: Using a black-white coloring to find a MST.

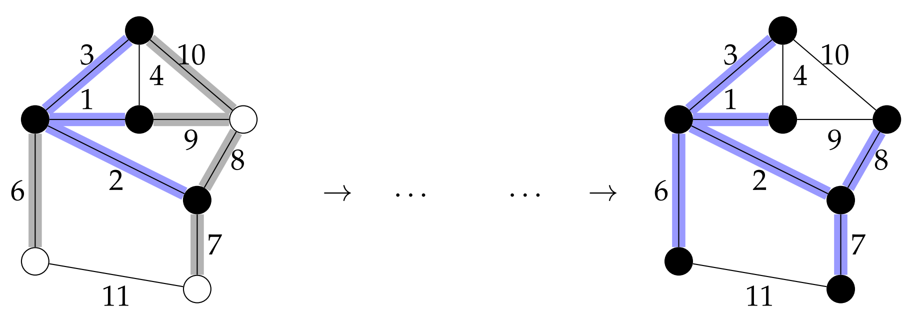

Figure 11: Resulting MST (blue).

Finally, we will prove these properties.

Proof of Cycle Property. We will prove the cycle property by contradiction. Suppose we have graph $G = (V, E)$ and the MST of G, T^*. Suppose the maximum weight edge $e^* = (u, v)$ of some cycle is in the MST: $e^* \in T^*$. Let the cycle containing e^* be C.

Let $T = T^* - \{e^*\}$. Since T^* is a spanning tree, e^* is a cut edge. Thus, T has two connected components. Because e^* was part of cycle C, there is a path connecting u and v in G not containing e^*. This path must contain an edge with endpoints in different connected components of T. Let such an edge be f. Note that $\boldsymbol{w}(f) < \boldsymbol{w}(e^*)$ because f and e^* are in cycle C and e^* is the edge with maximum weight in C. Now let $T = T^* - \{e^*\} + \{f\}$. T is connected, as f created a path between the two connected components. Additionally, T has $n - 1$ edges. Therefore T is a valid spanning tree. Then, we want to consider the weight of T. We have:

$$\boldsymbol{weight}(T) = \boldsymbol{weight}(T^*) - \boldsymbol{w}(e^*) + \boldsymbol{w}(f) < \boldsymbol{weight}(T^*).$$

Thus, T is a spanning tree of G with smaller weight than T^*, the minimum spanning tree of G. This is a contradiction. □

Proof of Cut Property. We will prove the cut property in the case that all edge weights are distinct. We proceed with a proof by contradiction. Suppose we have graph $G = (V, E)$, the MST of G, T^*, and a black-white coloring of G such that the minimum weight gray edge $e^* = (u, v) \in E$ of the is not in the MST: $e^* \notin T^*$.

Let $T = T^* + \{e^*\}$. By Theorem 4, adding e^* creates a cycle in T. An example of the creation of such a cycle can be seen in Figure 12. Note that u and v are both contained in this cycle since there are two disjoint paths connecting u and v, one inT^* and the other along e^*. Because e^* is a gray edge, u and v are different colors. Thus, the path from u to v in T^* must contain some gray edge (see an example in Figure 3). Let such an edge be f. Since e^* is the minimum weight gray edge, $\boldsymbol{w}(f) > \boldsymbol{w}(e^*)$.

Let $T = T^* + \{e^*\} - \{f\}$. e^* was not previously in T^*; if it were T^* would have contained a cycle. We claim that T is a spanning tree. T contains exactly $n - 1$ edges, as we simply swapped an edge in T^* for a different edge, and T contains every vertex. Additionally, T is acyclic, as we created and then broke a cycle containing u, v. Together, by Theorem 6 this implies T is connected. Since T is connected and uses $n - 1$ edges from G, it is in fact a spanning tree of G. Then, we want to consider the weight of T. We have:

$$\boldsymbol{weight}(T) = \boldsymbol{weight}(T^*) - \boldsymbol{w}(f) + \boldsymbol{w}(e^*) < \boldsymbol{weight}(T^*).$$

Thus, T is a spanning tree of G with smaller weight than T^*, the minimum spanning tree of T. This is a contradiction. □

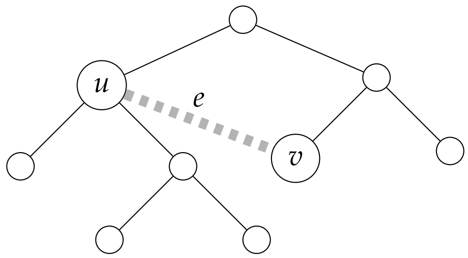

Figure 12: Tree T and min weight gray edge $e = (u, v)$ dashed.

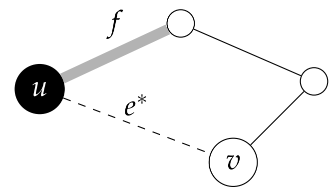

Figure 13: Cycle containing e^*.

Definition 8. *A cycle created by adding a single edge to a spanning tree is known as a* fundamental cycle (circuit).

Observe that the cycle we created in the proof of the cut property was a fundamental cycle.

4 Summary

In this lecture, we proved a relationship between connectivity and the number of edges in a graph. We introduced acyclic graphs, and proved properties pertaining to forests and trees. Finally, we defined spanning trees and considered finding the minimum spanning tree, a well-studied problem in computer science. We concluded by proving properties that are useful in algorithm design for this problem.